

# A MILP Scheduling Model for Multi-stage Batch Plants

Georgios M. Kopanos, Luis Puigjaner

*Universitat Politècnica de Catalunya - ETSEIB, Diagonal, 647, E-08028, Barcelona, Spain, E-mail: [luis.puigjaner@upc.edu](mailto:luis.puigjaner@upc.edu)*

## Abstract

In the current work, a new precedence-based mixed integer linear programming (MILP) scheduling framework, based on a continuous-time representation, is developed for the scheduling in multi-stage batch plants. Advantages and special features of the proposed scheduling model are highlighted through several instances of two base case studies. Results are analyzed and further criticized towards future work.

**Keywords:** scheduling, batch plants, multi-stage operations, MILP

## 1. Introduction

Multi-stage operations are found in a large number of industrial applications. The main features of multi-stage operations are the intermediate products storage strategy, such as zero wait (ZW), no intermediate storage (NIS), unlimited intermediate storage (UIS), and finite intermediate storage (FIS). Share resource constraints and sequence-dependent setup times are also of great importance since they complicate the problem (especially the latter ones). Neglecting these multi-stage operations characteristics leads to poor modelling of the real industrial process resulting to poor solutions once implemented.

In the PSE community a plethora of scheduling frameworks can be found. Among them continuous time representation strategies based on the precedence relationships between batches to be processed have been developed to deal with scheduling problems. Model variables and constraints enforcing the sequential use of shared resources are explicitly employed in these formulations. As a result, it is claimed that changeover issues can be treated in a straightforward manner. The three different precedence-based approaches that can be found in the literature, namely, are:

- i) the immediate precedence<sub>2</sub>,
- ii) the unit-specific immediate precedence<sub>3</sub>, and i
- iii) the general precedence<sub>4</sub>.

Immediate precedence explores the relation between each pair of consecutive orders in the production schedule time horizon without counting if the orders are assigned or not into the same unit. Unit-specific immediate precedence is based on immediate precedence concept. The difference is that it takes into account only the immediate precedence of the orders that are assigned to the same processing unit. General precedence (GP) generalizes the precedence concept by exploring the precedence relations of every batch regarding all the remaining batches and not only the immediate predecessor. The computational effort of this approach is significantly lower comparing it with the ones of the other two approaches.

Nevertheless, GP scheme appears some model representation drawbacks that may moderate its implementation in industrial practice. First of all, it cannot explicitly cope with sequence-dependent setup issues, such as times and costs. In other words, if changeover issues are the optimization objective function, or a part of it, GP cannot be used to tackle the addressed problem. To continue with, GP is inappropriate for solving scheduling problems wherein some products sub-sequences are forbidden. Finally,

even timing incoherencies resulting to myopic optimal solutions can be observed in some cases.<sup>5</sup>

## 2. Problem statement

In this work, the scheduling problem in multi-stage multiproduct batch plants with different processing units in parallel is addressed (see Fig.1). Batch-stage to unit assignment and batch sequencing in every processing stage meeting a production goal constitutes the under study scheduling problem.

The main problem characteristics and proposed model assumptions include:

- An equipment unit cannot process more than one batch at a time.
- Non-preemptive operation mode is assumed.
- Processing times, unit setup times and changeovers are deterministic.
- Unforeseen events are not appeared during the scheduling time horizon.
- Order driven demand pattern.
- Batch sizes are known a priori.

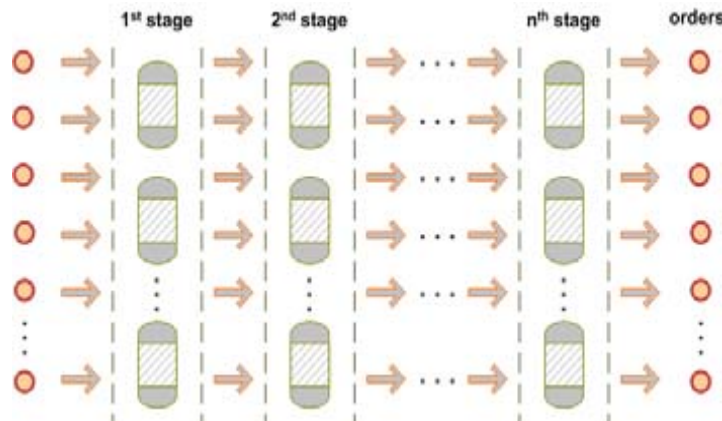


Figure 1. Multi-stage process scheme.

## 3. Mathematical formulation

The proposed precedence-based MILP model is based on a continuous-time domain representation. The main concept of the proposed model aims at combining the advantages that general precedence and immediate precedence frameworks appear, resulting to a new hybrid precedence-based formulation.

### 3.1. Allocation constraint

Every stage  $s$  of each order  $i$  can be assigned to at most one processing unit  $j$ :

$$\sum_{j \in ISJ_i} Y_{isj} \leq 1 \quad \forall i \in I^s, s \quad (1)$$

### 3.2. Timing constraints

The completion time of the first stage of product  $i$  has to be greater than its processing and its setup time plus the necessary changeover time,  $sd_{i'ij}$ , from the previous product  $i'$ , when products are consecutive into the same unit.

$$C_{is} \geq \sum_{j \in ISJ_i} (max[ru_j, ro_i] + pt_{isj} + su_{ij})Y_{isj} + \sum_{j \in ISJ_{i'}} \sum_{i' \in I^s} sd_{i'ij}Seq_{i'ij} \quad \forall i \in I^s, s = 1 \quad (2)$$

The binary variable,  $Seq_{i'ij}$ , becomes one when product  $i'$  is processed exactly before of product  $i$ , while both are allocated to the same unit; otherwise is set to zero.  $Seq_{i'ij}$  assess the unit-specific immediate precedence of two orders. To go on, the timing of the remaining stages is given by the next expression:

$$C_{is} \geq \sum_{j \in ISJ_i} (max[ru_j, ro_i] + pt_{isj} + su_{ij})Y_{isj} + \sum_{j \in ISJ_{i'}} \sum_{i' \in I^s} sd_{i'ij}Seq_{i'ij} \quad \forall i \in I^s, s = 1 \quad (3)$$

In eq.(3),  $tr_{s-1s}$  corresponds to the transfer time between two sequential stages of a particular product  $i$  while  $Stor_{is-1}$  stands for the time that the stage  $s-1$  of a product  $i$  is stored before proceeding to the following processing stage  $s$ . In ZW storage police  $Stor_{is-1}$  is set to zero. In NIS is left free. In order to model storage policies like UIS and FIS, equations found in the literature can be added.

### 3.3. Sequencing-timing constraints

Binary variables  $X_{i'ij}$  and  $Seq_{i'ij}$  follow the unit-specific general precedence and the unit-specific immediate precedence notion, respectively. Roughly speaking,  $X_{i'ij}$  is 1 when product  $i$  is processed before product  $i'$  in the same unit  $j$ , in contrast with  $Seq_{i'ij}$  that is 1 when order  $i$  is the predecessor of order  $i'$  in unit  $j$ .

$$C_{is} + sd_{ii'j}Seq_{ii'j} \leq C_{i's} - pt_{i'sj} - su_{i'j} + M(1 - X_{ii'j}) \quad \forall i \in I^s, i' \in I^s, i' \neq i, s, j \in (ISJ_i \cap ISJ_{i'}) \quad (4)$$

Eq.(4) states that the starting time of an order  $i'$  is greater than the completion time of whichever order  $i$  processed beforehand. The binary  $Seq_{i'ij}$  activates only the changeover times between consecutive orders, thus assessing sequencedependent issues explicitly; and not implicitly as general precedence does.

### 3.4. Sequencing-allocation constraints

Eqs.(5)-(6) are needed in order to formulate the unit-specific general precedence concept of the proposed model.

$$Y_{isj} + Y_{i'sj} \leq 1 + X_{ii'j} + X_{i'ij} \quad \forall i \in I^s, i' \in I^s, i' \neq i, s, j \in (ISJ_i \cap ISJ_{i'}) \quad (5)$$

$$2(X_{ii'j} + X_{i'ij}) \leq Y_{isj} + Y_{i'sj} \quad \forall i \in I^s, i' \in I^s, i' \neq i, s, j \in (ISJ_i \cap ISJ_{i'}) \quad (6)$$

These equations state that when two orders are allocated to the same unit, i.e.  $Y_{isj} = Y_{i'sj} = 1$ , only one of the two binary variables  $X_{ij}$  and  $X_{i'j}$  will be one. If the two orders are not allocated to the same unit then  $X_{ij} = X_{i'j} = 0$ .

### 3.5. Assessing consecutiveness through general precedence

Obviously, two orders  $i$  and  $i'$  are consecutive only in the case that the binary variable  $X_{iij}=1$  and, moreover, when there is no other order,  $i''$ , between them.

$$Pos_{iij} = \sum_{i'' \neq [i, i']} (X_{i i'' j} - X_{i' i'' j}) + M(1 - X_{iij}) \quad \forall i, i' \neq i, j \in (JI_i \cap JI_{i'}) \quad (7)$$

$$Pos_{iij} + Seq_{iij} \geq 1 \quad \forall i, i' \neq i, j \in (JI_i \cap JI_{i'}) \quad (8)$$

In eq.(7), the auxiliary variable  $Pos_{iij}$  is set to zero *if and only* two products  $i$  and  $i'$  are consecutive and are assigned to the same unit. In other words, when order  $i$  is processed before order  $i'$  in unit  $j$ , i.e.  $X_{iij}=1$ , and the summation term in eq.(7) is zero, i.e. there is no other order  $i''$  between them, then the two orders are consecutive. In any other case,  $Pos_{iij}$  gets a value different than zero. Eq.(8) forces the  $Seq_{iij}$  variable to be 1 whenever  $Pos_{iij}=0$ ; in any other case,  $Seq_{iij}$  is set to zero. The 1 in the right-hand side of eq.(8) can be substituted by  $X_{iij}$ . It has found that in some instances reduces the computational time.

## 4. Case studies

A simplified (regarding the number of products) version of an industrial case study of a multi-stage pharmaceuticals batch plant is considered. Products are processed to 5 or 6 stages and changeover times (in same stages higher than the processing times) are present. Four different instances (I.1-I.4) of this case study have been solved. Five products are scheduled in all cases but case I.2; wherein six products are considered. Makespan minimization is the optimization goal in I.1-I.2, changeovers costs minimization in I.3, and operating plus changeovers costs minimization in I.4. ZW policy is applied. Computational results can be found Table 1. A 10-minute time limit has been imposed. Fig.2 shows the optimal schedule of I.2 problem; note that only USGP reached optimal solution.

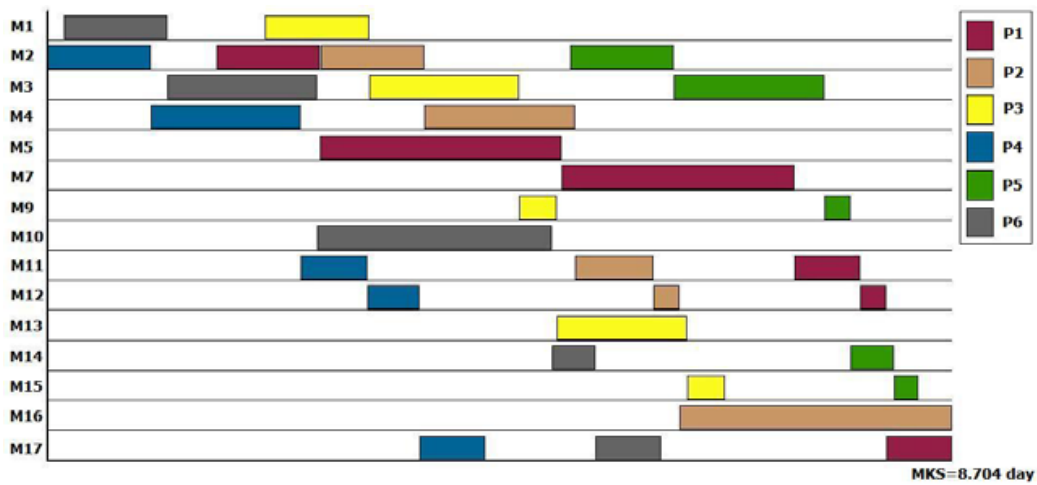


Figure 2. Optimal schedule for I.2 problem instance (USGP optimal solution).

Afterwards, a modified case study<sup>6</sup> found in the literature is addressed. Six 3-stage products are to be scheduled under UIS policy. Three different instances (II.1-II.3) of this case have been solved. Final and intermediate inventory minimization is the objective in cases II.1-II.2. In case II.3 the minimization of both total inventory and changeover costs is desired. In II.1 example changeover times are present while in II.2 they are set to zero, in order to show how the problem difficulty decrease by not considering changeovers. GP hasn't reached the optimal solution in II.1 case in the 5-minute time limit. Computational results are included to Table 1. Fig.3 depicts the optimal schedule for II.3 problem.

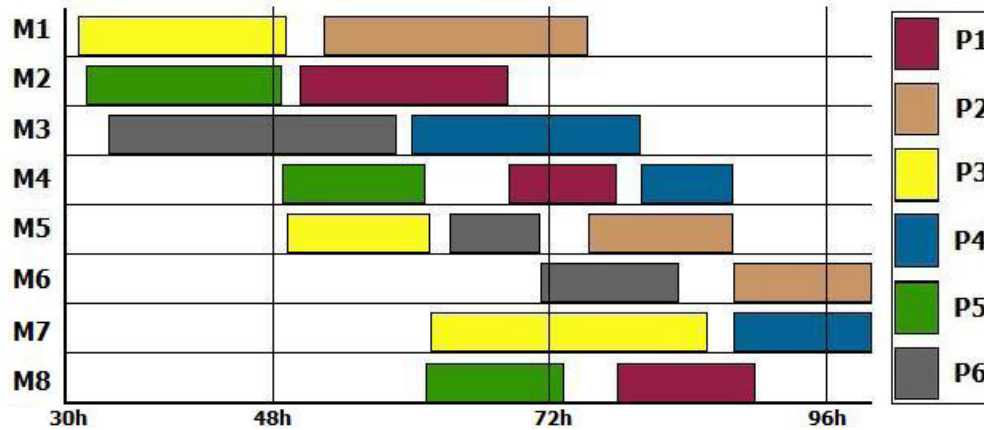


Figure 3. Optimal schedule for II.3 problem instance.

Problem	Model	Obj. funct.	N. of eqns	Bin. vars	Cont. vars	Nodes	CPU (s)
I.1	GP	8.506	266	166	26	40	0.365
I.1	USGP	8.506	1,106	271	485	334	0.575
I.2	GP	(1.83%) 8.740	391	236	31	795,979	> 600
I.2	USGP	8.704	2,011	398	730	91,504	406.680
I.3	USGP	29.583	1,102	271	485	750,288	559.100
I.4	USGP	905.956	1,107	271	486	57,885	43.160
II.1	GP	(48.7%) 32.01	288	168	42	1,404,379	> 300
II.1	USGP	31.72	1,248	288	522	47,948	103.81
II.2	GP	26.10	288	168	42	2,116,139	452.15
II.2	USGP	26.10	1,248	288	522	14,068	22.65
II.3	USGP	48.39	1,248	288	522	27,452	66.55

\*Solved in GAMS (CPLEX 11.0) in a Dell Inspiron 1526, 2 GHz with 2 GB RAM

## 5. Final discussion and future work

Taking into consideration that changeover issues are usually a crucial part of the optimization goal in a large number of industries, the current work has been focused on the development of an efficient MILP model appropriate for tackling this kind of problems. In all complicated cases, the proposed model performance overwhelms that of GP; in spite of its bigger model size. USGP has been found to be much faster even in cases, e.g. I.2, II.1 and II.2, where GP was expected to perform better, mainly because of GP's small model size. Future work will be focused on developing decomposition strategies in order to reduce even more the computational burden that is required to solve large-scale industrial problems, such the complete pharmaceuticals case is.

## 6. Acknowledgements

Financial support received from the Spanish Ministry of Education and Science (FPU grants) is fully appreciated. Authors would like to express their gratitude to Carlos Méndez for providing them with the pharmaceuticals case study.

## 7. References

- [1] C. Méndez, J. Cerdá, I.E. Grossmann, I. Harjunkski, M. Fahl, 2006, *Comput. Chem. Eng.* 30, 913-946.
- [2] C. Méndez, G. Henning, J. Cerdá, 2000, *Comput. Chem. Eng.*, 24, 2223-2245.
- [3] J. Cerdá, G. Henning, I.E. Grossmann, 1997, *Ind. Eng. Chem. Res.*, 36, 1695-1707.
- [4] C. Méndez, J. Cerdá, 2003, *Comput. Chem. Eng.*, 27, 1247-1259.
- [5] G.M. Kopanos, L. Puigjaner, *AIChE Annual Meeting 2008*, 554d.
- [6] J.M. Pinto, I.E. Grossmann, 1997, *Comput. Chem. Eng.*, 21, 8, 801-818